



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/916,663	07/26/2001	Alain Sarraf	US20 01 0040	3478

7590 10/01/2004

Paul D. Greeley, Esq.  
Ohlandt, Greeley, Ruggiero & Perle, L.L.P.  
10th Floor  
One Landmark Square  
Stamford, CT 06901-2682

EXAMINER

MOFIZ, APU M

ART UNIT	PAPER NUMBER
----------	--------------

2175

DATE MAILED: 10/01/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

**Office Action Summary**

Application No.

09/916,663

Applicant(s)

SARRAF ET AL.

Examiner

Apu M Mofiz

Art Unit

2175

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 18 June 2004.  
2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.  
3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-12 is/are pending in the application.  
4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.  
5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.  
6) ☒ Claim(s) 1-12 is/are rejected.  
7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.  
8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

DIANE D. MIZRAHI  
PRIMARY PATENT EXAMINER  
TECHNOLOGY CENTER 2100

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.  
10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).  
11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☒ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.  
3. ☒ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)  
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)  
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_.  
4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date \_\_\_\_\_.  
5) ☐ Notice of Informal Patent Application (PTO-152)  
6) ☐ Other: \_\_\_\_\_.

## DETAILED ACTION

### *Response to Arguments*

1. Applicant's arguments submitted on 06/16/2004 with respect to claims 1-12 have been reconsidered but are not deemed persuasive for the reasons set forth below.

Applicant argues (on page 8, under REMARKS section) that the references do not teach or suggest (a) creating on a server, a list of required structs based on a requirement of the client, (b) sending the list from the server to the client, and (c) processing, by the client, data represented by the list.

Examiner respectfully disagrees. Nodushani teaches (a) creating on a server, a required struct based on a requirement of the client, ( i.e.

```
//module MTOAMP1 {  
    //  
    ...  
    //structure definitions  
    ...  
    // Attributes for the T1 interface  
    //  
    struct T1Attributes {  
        unsigned short ifIndex;  
        Boolean primary;  
        ...  
    }  
}
```

```
};  
  
...  
  
interface T1{  
  
...  
  
    T1Attributes getAttributes ();  
  
...  
  
};  
  
...
```

};" the preceding text/code excerpts clearly indicates that T1Attributes is a definition of a struct and interface T1 has a method getAttributes() which returns T1Attributes to the client. In a CORBA environment, an interface is a client stub that works as a proxy for the client. Here the client needs the T1Attributes from the server and invokes the getAttributes() method and the server creates T1Attributes and returns them to the client. It requires no further explanation.) (col 46; lines 15-24; col 50, lines 15-33) (b) sending the struct from the server to the client, (col 46; lines 15-24; col 50, lines 15-33) and (c) processing, by the client, data represented by the struct (i.e. the returned T1Attributes from the server is for the client to process or whatever it wants to do with these.) (col 46; lines 15-24; col 50, lines 15-33).

Rodriguez teaches that the client requests a list (i.e. a sequence of Markers/ structs) of structs from the server (page 5).

### ***Claim Rejections - 35 USC § 103***

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the

Art Unit: 2175

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

3. Claims 1-12 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nodoushani et al. (U.S. Patent No. 6,563,816 and Nodoushani hereinafter) in view of Rodriguez et al. ("A CORBA server for the Radiation Hybrid DataBase", 1997, pages 1-8 and Rodriguez hereinafter).

As to claims 1,6,7,8,9 and 11, Nodoushani teaches a method of operating a communication network (i.e. the end-to-end system) (Fig. 28; col 38, line 65) with a network management system (i.e. *"The CPA 42 further includes a home LAN manager 389, an SNMP (simple network management protocol) agent 390 and general UNIX administration utilities 388. Multiple HLHs 20 are managed from the CPA 42 through channels 362C to HL agents 380 to support all provisioning, maintenance and testing functions."*) (col 39, lines 12-17), wherein said communication network (Fig. 28; col 38, line 65) comprises an object (i.e. CORBA-based managed object) (col 2, lines 45-54; col 39, lines 24-29) on a server (i.e. *"According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which can manage the plural CORBA-based managed objects directly." ... "A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396."*) (col 2, lines 45-54; col 39, lines 24-29), and wherein said network management system (col 39, lines 12-17) is located on said server (col 2, lines 45-54; col 39, lines 24-29) and can be accessed by a client (i.e. *"Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-*

3, 397-4." ... *"There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE TL1 protocol, if desired to inter-operate with OSSs."* ) (col 40, lines 10-27), wherein said server and said client transmit data over said communication network in accordance with a Common Object Request Broker Architecture (CORBA) (col 2, lines 45-54; col 39, lines 24-29), and wherein said method comprises: accessing by said client, a descriptor for said object, wherein said descriptor is represented by a struct ( i.e.

```
//module MTOAMP1 {  
  
    //  
  
    ...  
  
    //structure definitions  
  
    ...  
  
    // Attributes for the T1 interface  
  
    //  
  
        struct T1Attributes {  
  
            unsigned short ifIndex;  
  
            Boolean primary;  
  
            ...  
  
        };  
  
    ...  
  
    interface T1{  
  
        ...  
  
        T1Attributes getAttributes ();  
  
        ...  
  
    };  
  
    ...
```

};" ) (col 46; lines 15-24; col 50, lines 15-33) that includes an attribute but no operation (i.e. the structure by definition does not contain any operation/method as in the prior art) (col 46; lines 15-24; col 50, lines 15-33); creating on said server, required struct based on a requirement of said client ( i.e.

```
//module MTOAMP1 {  
  
    //  
  
    ...  
  
    //structure definitions  
  
    ...  
  
    // Attributes for the T1 interface  
  
    //  
  
        struct T1Attributes {  
  
            unsigned short ifIndex;  
  
            Boolean primary;  
  
            ...  
  
        };  
  
    ...  
  
    interface T1{  
  
        ...  
  
        T1Attributes getAttributes ();  
  
        ...  
  
    };  
  
    ...  
  
};"
```

};" the preceding text/code excerpts clearly indicates that T1Attributes is a definition of a struct and interface T1 has a method getAttributes() which returns T1Attributes to the client. In a CORBA environment, an interface is a client stub that works as a proxy for the client. Here the client needs the T1Attributes from the server

Art Unit: 2175

and invokes the `getAttributes()` method and the server creates `T1Attributes` and returns them to the client. It requires no further explanation.) (col 46; lines 15-24; col 50, lines 15-33) ; sending said required struct from said server to said client (col 46; lines 15-24; col 50, lines 15-33); and processing, by said client, data represented by said struct (i.e. the returned `T1Attributes` from the server is for the client to be processed or whatever it wants to do with these.) (col 46; lines 15-24; col 50, lines 15-33).

Nodoushani does not explicitly teach an object stored in a database on a server and Nodoushani does not teach that the client requests a list of structs from the server.

Rodriguez teaches an object stored in a database on a server (page 4) and Rodriguez also teaches that the client requests a list (i.e. a sequence of `Markers/ structs`) of structs from the server (page 5).

It would have been obvious to a person of ordinary skill in the art at the time of Applicant's invention to modify the teachings of Nodoushani with the teachings of Rodriguez to include an object stored in a database on a server and the client requesting a list of structs from the server with the motivation to provide an application of CORBA technology to support external access to the database (Rodriguez, page 4).

As to claim 2, Nodoushani teaches using a definition for an interface (col 50, lines 60-65) that only includes an operation but not necessarily any attribute (i.e.

```
"Interface RegisterClient {  
    void signOn (in TrapFieldier client);  
    ...  
};" (col 50, lines 60-65).
```



As to claim 3, Nodoushani teaches including a reference to said struct in said interface ( i.e.

```
//module MTOAMP1 {  
    //  
    ...  
    //structure definitions  
    ...  
    // Attributes for the T1 interface  
    //  
    struct T1Attributes {  
        unsigned short ifIndex;  
        Boolean primary;  
        ...  
    };  
    ...  
    interface T1{  
        ...  
        T1Attributes getAttributes ();  
        ...  
    };  
    ...  
};
```

};") (i.e. the above code excerpts show that the struct for the "T1Attributes" is a descriptor for a corresponding interface with the name "T1". The struct only comprises the attributes of the T1, but not its operation. The operation of the T1 i.e. the getAttributes operation is contained in the definition of the "T1". The attributes are referenced by the client via the T1Attributes, which is defined in the described struct descriptor. The attributes are therefore accessible via the T1Attributes without the need for a T1. Another interface/client needs a reference to the

T1Attributes to access/store the same attribute information as that which is defined in a known interface and is essentially depended on the particular need of the client application) (col 46; lines 15-24; col 50, lines 15-33).

As to claim 4, Nodoushani teaches that an instruction from the client (i.e. "Fig. 29 shows a configuration for CORBA-based flow through provisioning with the CORBA server 386 of CPA 42 and the aforementioned OSSs 397-1, 397-2, 397-3, 397-4." ... "There are several advantages of the CORBA-based configuration. For example, by basing the internal management scheme on CORBA, instead of raw memory, it is very easy to add new protocols as CORBA clients, such as BelCORE TL1 protocol, if desired to inter-operate with OSSs." ) (col 40, lines 10-27) which is directed to the struct ( i.e.

```
//module MTOAMP1 {  
    //  
    ...  
    //structure definitions  
    ...  
    // Attributes for the T1 interface  
    //  
    struct T1Attributes {  
        unsigned short ifIndex;  
        Boolean primary;  
        ...  
    };  
    ...  
};
```

};" ) (col 46; lines 15-24; col 50, lines 15-33) on the server (i.e. "According to an aspect of the invention, a network element includes a Common Object Request Broker Architecture (CORBA)-based server, CORBA-based managed objects accessible by the CORBA-based server and a CORBA-based applications programming interface (API). The CORBA-based API is coupled to an external operations support system which

*can manage the plural CORBA-based managed objects directly." ... "A CORBA server 386 provides the "glue" for all management applications with information regarding provisioning 382A, inventory 394, status 384A and alarms 396."*) (col 2, lines 45-54; col 39, lines 24-29), results in sending data ( i.e.

```
"/module MTOAMP1 {
```

```
...
```

```
interface TI{
```

```
...
```

```
TIAttributes getAttributes ();
```

```
...
```

```
};
```

```
...
```

```
};" ) (col 46; lines 15-24; col 50, lines 15-33) from the server (col 2, lines 45-54; col 39, lines 24-29) to the client (col 40, lines 10-27).
```

As to claim 5, Nodoushani teaches not storing the data sent to the client on the server (i.e. "*TIAttributes getAttributes ();*". The code excerpts show that the client invokes the *getAttributes* method and the server sends the attributes to the client in a struct but does not store them in the server.) (col 46; lines 15-24; col 50, lines 15-33)).

As to claim 10, Nodoushani teaches that the said computer program (col 46; lines 15-24; col 50, lines 15-33) is stored on a data carrier (i.e. a computer or any other processor) (Fig. 28).

As to claim 12, Nodoushani teaches that the said computer program product (col 46; lines 15-24; col 50, lines 15-33) is stored on a data carrier (i.e. a computer or any other processor) (Fig. 28).

### ***Conclusion***

4. **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

### ***Points of Contact***

5. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Apu M. Mofiz whose telephone number is (703) 605-4240. The examiner can normally be reached on Monday – Thursday 8:00 A.M. to 4:30 P.M.

Art Unit: 2175

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Dov Popovici can be reached at (703) 305-3830. The fax numbers for the group is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Apu M. Mofiz  
Patent Examiner  
Technology Center 2100

September 28, 2004

  
DIANE O. MZRAHI  
PRIMARY PATENT EXAMINER  
TECHNOLOGY CENTER 2100